USB Data Logger

Prof Ioannis Lestas
CUED
icl20@cam.ac.uk

Demonstrators:
Rachel Hudson
Giovanni Pugliese Carratelli
Parfait Lutundula
*CUED*

For each student

- SF4 handout 1
- SF4 handout 2
- SF4 handout 3 (introductory exercises)

Also available on moodle + additional reading material

develop a modern embedded data logging system

1   Hardware – design analogue circuitry for taking input signal

2   Firmware – control microcontroller unit and handle input

3   Communications/Software – protocol of data
transfer over USB to PC, PC user interface

4   Product Identity – datasheet to advertise product,
choose how to spend £15 budget

5   Experience the dynamics/challenges of research project work
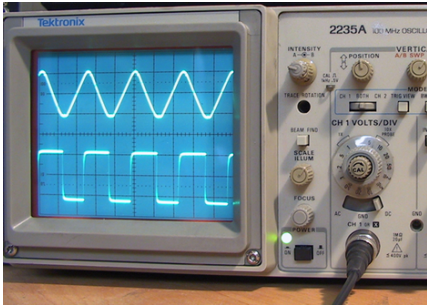
Oscilloscope
- Fully featured
  e.g. FFT, Dual Trace

Radio
- Recording
- Station, volume control
- Autotune

Healthcare monitors
- Oximeter
- Heart Rate Monitor

Music Accessories
- Tuner
- Effects amp/pedal

- Recording + playback
- Mixing Desks
- Turntables

Weather station
- Temperature
- Wind speed
- Humidity

SF4 timeline

10/05	Fri 11-1pm (introduction)
14/05	Tue 9-11am
14/05	Tue 2-4pm
17/05	Fri 11-1pm
21/05	Tue 9-11am (**Interim Report**)
21/05	Tue 2-4pm
24/05	Fri 11-1pm (**rep. feedback**)
28/05	Tue 9-11am
28/05	Tue 2-4pm
31/05	Fri 11-1pm
04/06	Tue 9-11am
04/06	Tue 2-4pm (**Demonstration**)
07/06	Fri, 4pm, submit in **final report (moodle)**

✓ work in pairs (ensure complementary knowledge)

✓ timetabled sessions are
  supported by 1-2 demonstrators,
  compulsory,
    attendance sheets are used

✓ additional: EIETL access hours:
    8-4pm weekdays
  check PC terminal availability
  (overall time budget ~20h/per week)

✓ keep back-ups of computer files/reports

	Milestone	When and what happens	What is required
1	Specification and design – **Interim Report**	Date: 21 May 2023 by 11am Report feedback for 10 mins, in pairs, on 24 May Worth: **10 marks**	**1 set of paperwork per pair** containing (2 pages, excl. appendices): analogue circuit design, parts list for purchasing, block diagrams for comms., firmware, software design. Plus: **Who's doing which bit?**
2	**Demonstration**	Date: 4 June 2023 (2-4pm) Students (in pairs) demonstrate their system with the assessor for 15 minutes Worth: **30 marks**	Working hardware; firmware and software (so it can be copied during the demo)
3	**Final Report**	Date: 7 June 2023, 4pm (moodle) Students must **individually** write up and hand in their report. Each will be assessed on its own merits. Worth: **40 marks**	**1 marketing datasheet** from each student [1 piece of A4 (2-sides) max] **1 independent report** from each student (10 sides of A4 max, excluding appendices, as outlined in THIRD YEAR PROJECT GUIDE*; add interim report as appendix). **Signed declaration** that student is submitting own work.

Each project will be evaluated based on:

✓design creativity and functionality

✓2-way communications between MCU and computer

✓analogue-digital conversion over multiple channels and/or kHz sampling rate
         (or appropriate alternative)

✓analogue circuitry including appropriate conditioning
  (eg. anti-aliasing, automatic gain control, DC offset removal etc if appropriate)

✓level of signal processing (eg. Fourier transforms, scaling, averaging etc.)

✓Usability of software interface

✓robust demonstration of whole system

Clear presentation and effective communication of system design and its features in the interim and final reports as well as in the datasheet.

➢ For non-attendance at compulsory timetabled sessions, the penalty is 1 mark per hour or part hour missed.

➢For late submission of interim reports, the penalty is 3 marks per day.

➢**No reports will be accepted after the submission date for the final report.** (computer problems are not acceptable reasons)

➢Feedback will be given on each report, but the marks will not be disclosed.

➢If work is disrupted by illness, project leader, tutor and Dr Alex White must be notified immediately.

➢Max. total length of all reports taken together must not exceed 13 A4 sides, excl. appendices (Interim: 2 pages, Datasheet: 1 page, Final Report: 10 pages)

➢Each report requires a signed statement that you are submitting your own work (Note: "No need to reinvent wheel", BUT you must cite previous work and majority of final report must be your own work)

➢Project moderators will award prizes to top students

Project based around Arduino microcontroller
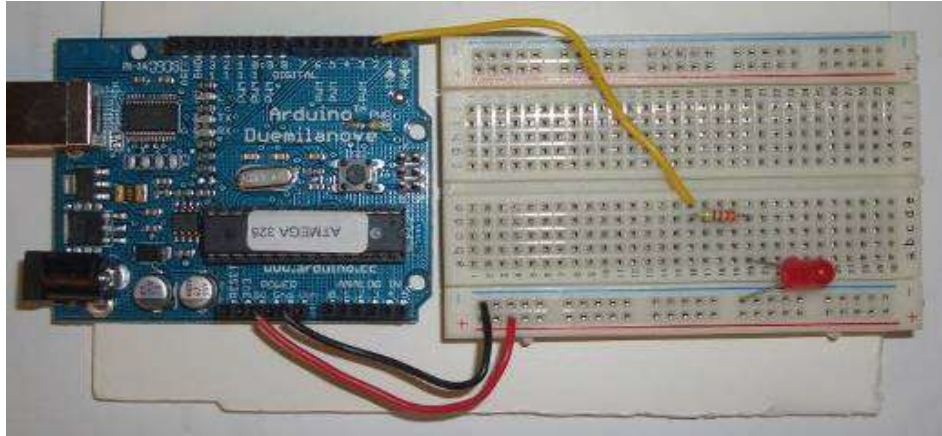
✓ USB bus powered

✓ Programmed and debugged via USB

The microcontroller board is supplied, additional circuits can be constructed on the breadboard

A budget of £15 is provided for purchasing additional components for your circuit (excludes resistors and capacitors from EIETL)
– order from "Onecall" through EIETL technicians, once approved by demonstrators

Go through Introductory Handout

Familiarise yourself with software & components

Familiarise yourself with the arduino board
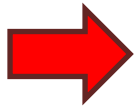
Future sessions: create your own design and implement it. Handouts 2 & 3 provide some advice and structure.

Plan your time carefully

- Components can be ordered before 21st May (Interim report)
- Start firmware/software development before your components arrive
- Fault-finding and debugging might take a lot of the time later on

Make use of the handouts

Start simple

- Come up with an achievable concept
- Start by getting a basic system working and then build on that
- Add advanced features later

Ask demonstrators for help and advice

Hardware:

- Design your circuit carefully
- Discuss with demonstrators
- Check datasheets thoroughly before ordering components
- Build a neat circuit
- Test each sub-system separately before connecting the next one
- Don't make your entire circuit before testing it
- 5V and 3.3V supplies available from development board (USB bus powered)
- Use the fuses provided
- 20 mA at each I/O pin (> 40mA causes permanent fault) - if you need more current you'll need to provide your own power supply

SF4   Tips

Firmware and Software:
- Comment code
- Divide code up sensibly into functions
- Use separate C files as separate modules of code
- Make code readable
- Make small changes and then test them
- Don't write large blocks of code without testing each part
- Use the debugger to see what's going on in the firmware and diagnose problems
- Demonstrators can help you more quickly when code is well written
- Be careful writing firmware in C – as a very low level language it allows you to make mistakes that can be hard to debug
- To install new python libraries on the EIETL machines you need to follow the instruction in your handout (run a code provided on moodle)